

FIR Filter Implementation by Systolization using DA-based Decomposition

Shreeharsha K G, Rekha Bhandarkar

Department of Electronics and Communication Engineering Alva's Institute of Engineering and Technology,
Mijar – 574 225, Moodbidri, INDIA

shreeharshakg01@gmail.com

Rekha Bhandarkar

Department of Electronics and Communication Engineering

NMAM Institute of Technology, Nitte – 574 110

Karkala, INDIA

rekhabhandarkar@rediffmail.com

Abstract— In this paper we present 1D and 2D systolic Distributed Arithmetic (DA) based structures that are designed for the implementation of Finite Impulse Response (FIR) filters. The paper compares the 1D DA based systolic structure with 1D systolic DA based decomposition method. The filters are implemented on a Xilinx Virtex II Pro (XC2VP30) FPGA using HDL and system metrics like Area, Gate Count, Maximum Usable Frequency and Power consumption are estimated for different filter orders and address lengths. The 1D systolic decomposition structure is also compared with the existing system generator implementation of DA FIR.. Results for an exemplary implementation are presented.

Keywords— Distributed arithmetic (DA), Field Programmable Gate Arrays (FPGA), Finite Impulse Response (FIR) filter, systolic array.

I. INTRODUCTION

Finite Impulse Response (FIR) filters are one of the most common components of Digital Signal Processing (DSP) systems. FIR filtering is achieved by convolving the input data samples with the desired unit response of the filter. Since the complexity of implementation grows with the filter order and the precision of computation, real-time realization of these filters with desired level of accuracy is a challenging task. Several attempts have, therefore, been made to develop dedicated and reconfigurable architectures for realization of FIR filters in Application Specific Integrated Circuits (ASIC) and FPGA platforms. DA provides an approach for multiplier-less implementation of FIR filters where the filter coefficients are programmable. In other words, the same filter structure can be used for a different set of coefficients.

A systolic system consists of a set of interconnected cells, each capable of performing some simple operation. Because simple, regular communication and control structures have substantial advantages over complicated ones in design and implementation, cells in a systolic system are typically interconnected to form a systolic array or a systolic tree. Information in a systolic system flows between cells in a pipelined fashion, and communication with the outside world occurs only at the “boundary cells.” For example, in a systolic array, only those cells on the array boundaries may be I/O ports for the system[5]. The basic principle of a systolic

architecture, array in particular, is replacing a single Processing Element (PE) with an array of PEs or cells. Being able to use each input data item a number of times (and thus achieving high computation throughput with only modest memory bandwidth) is one of the advantages of the systolic approach. They have several attractive features such as simplicity, regularity and modularity of structure [2]. In addition, they also possess significant potential to yield high-throughput rate by exploiting high-level of concurrency using pipelining or parallel processing or both.

II. DISTRIBUTED ARITHMETIC

Distributed Arithmetic (DA) is an efficient method for computing inner products when one of the input vectors is fixed[4]. It uses look-up tables and accumulators instead of multipliers for computing inner products.

Let us consider the inner-product of two N-point vectors A and B given by Eq. (1) as,

$$C = \sum_{k=0}^{N-1} A_k B_k \quad (1)$$

where A is a constant vector, while B may change from time to time. Assuming L to be the word length, each component of B may be expressed in two's complement representation. The inner-product can be expressed in distributed form as shown below in Eq. (2):

$$C = -\sum_{k=0}^{N-1} A_k b_{k0} + \sum_{l=1}^{L-1} 2^{-l} \left[\sum_{k=0}^{N-1} A_k b_{kl} \right] \quad (2)$$

where b_{kl} denotes the l^{th} bit of B_k .

For simplicity, assuming the signal samples to be unsigned words of size L, the inner product given in Eq. (2) then can be expressed in a simpler form by Eq. (3) as,

$$C = \sum_{l=0}^{L-1} 2^{-l} \cdot C_l \quad (3)$$

According to the decomposition scheme proposed by Meher [1], when N is a composite number given by $N = PM$, (where P and M may be any two positive integers) one can map the index k into $(m + pM)$ for $m = 0, 1, \dots, M-1$ and $p = 0, 1, \dots, P-1$. Hence equation (3) can be expressed in the following form as Eq. (4a):

$$C = \sum_{l=0}^{L-1} 2^{-l} \left(\sum_{p=0}^{P-1} F(b_p)_{l,p} \right) \quad (4a)$$

where

$$(b_n)_{i,p} = \sum_{m=0}^{M-1} a(m + pM) \cdot (b_n(m + pM))_i \quad (4b)$$

for $i = 0, 1, \dots, L-1$ and $p = 0, 1, \dots, P-1$.

The bit vector $(b_n)_{i,p}$ in Eq. (4b) is used as address word for the lookup table and F is the memory-read operation.

III. 1-D SYSTOLIC ARRAY FOR FIR FILTERS

A linear array consisting of P number of PEs and an output cell is shown in Fig. 1 and the function of the PEs is described in Fig. 1(b)[3].

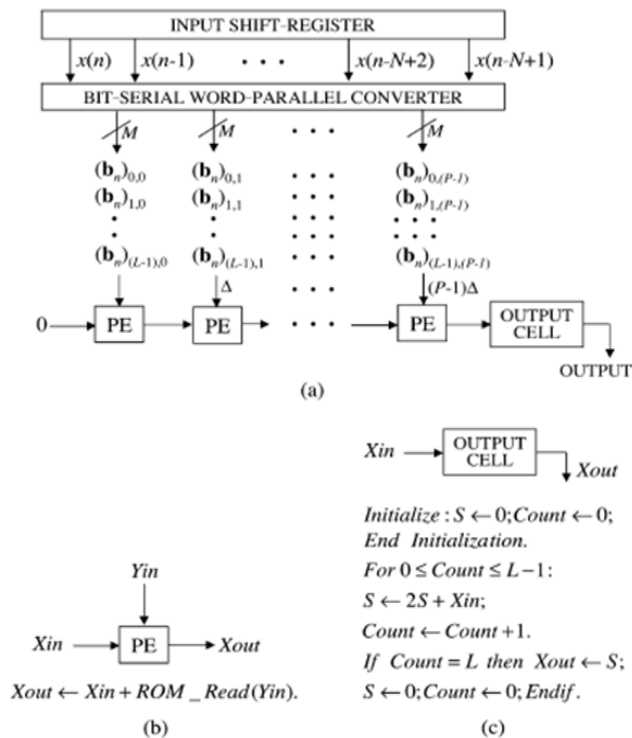


Figure. 1. The 1-D array for DA-based implementation of FIR filter: (a) Linear systolic array; (b) function of PE; and (c) function of output cell. delta stands for a unit delay.

The input sequence $\{x(n)\}$ is fed to a serial-in parallel-out input register where content of the register is serially right shifted by one position and transferred in parallel to the bit-serial word-parallel converter in every L cycles. The function of the output cell is shown in Fig. 1(c). After L cycles, it delivers a desired filter output. The structure will yield its first filter output $(L+P)$ cycles after the first input is fed to the first PE, while the successive output becomes available in every L cycles.

IV. IMPLEMENTATION

This section is concerned with the description of the implementation of the FIR filter based on conventional and systolic decomposition of DA-based computation.

TABLE I
PERFORMANCE OF THE PROPOSED FPGA IMPLEMENTATION OF THE 1-D SYSTOLIC DECOMPOSITION METHOD FOR FIR FILTER FOR WORD LENGTH $L = 8$

FILTER ORDER (N)	ADDRESS LENGTH (M)	AREA (SLICES)	FREQUENCY (MHz)	POWER CONSUMPTION (mW)	GATE COUNT
8	2	104	251.826	44	2120
	4	99	297.885	39	1854
	8	159	232.829	45	2236
16	2	205	175.623	42	3793
	4	187	284.738	40	3244
	8	329	197.981	53	4404
32	2	330	174.672	46	6639
	4	343	153.799	45	5794
	8	606	168.209	60	8489

From the data presented in Table 1 and from the Figures 2 to 5 it can be seen that for a given filter order N , the case for $M = 4$ yields the most area-time efficient architecture when compared to the case for $M = 2$ and 8. This can be explained by the fact that the increase in control logic and number of delay elements outweighs the gains made by reduction of LUT size for $M = 2$, while for $M = 8$, the memory requirement of LUTs is too high [1]. Frequency is also maximum for lower orders. Power consumption is the lowest.

TABLE II
COMPARISON OF THE 1-D SYSTOLIC CONVENTIONAL METHOD AND 1-D SYSTOLIC DECOMPOSITION METHOD WITH ADDRESS LENGTH $M = 4$ FOR FIR FILTER FOR WORD LENGTH $L = 8$

	1-D SYSTOLIC CONVENTIONAL METHOD			1-D SYSTOLIC DECOMPOSITION METHOD		
FILTER ORDER (N)	8	16	32	8	16	32
ADDRESS LENGTH (M)	9	17	33	4	4	4
AREA (SLICES)	124	307	644	99	187	343
FREQUENCY (MHz)	238.095	203.832	275.103	297.885	284.738	153.799
POWER CONSUMPTION (mW)	43	47	54	39	40	45
GATE COUNT	2066	4213	8543	1854	3244	5794

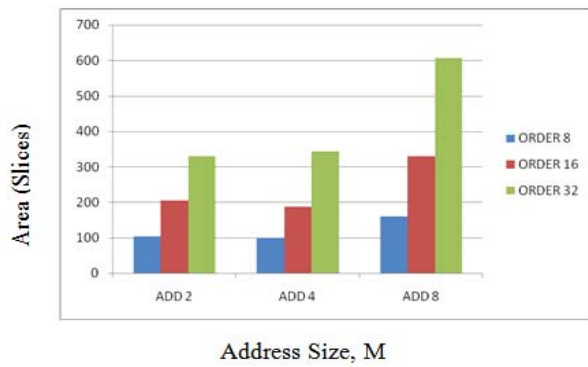


Figure 2. Plot of variation of Area with filter order for 1-D Decomposition method for $L = 8$.

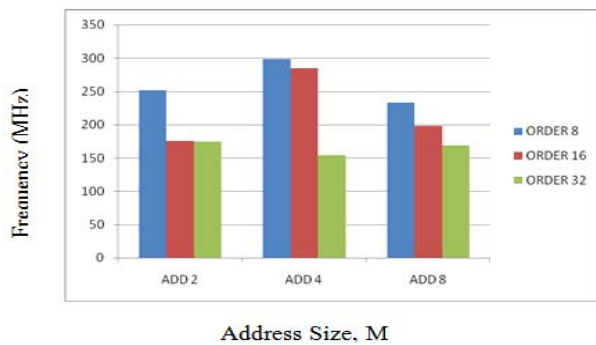


Figure 3. Plot of variation of Frequency with filter order for 1-D Decomposition method for $L = 8$.

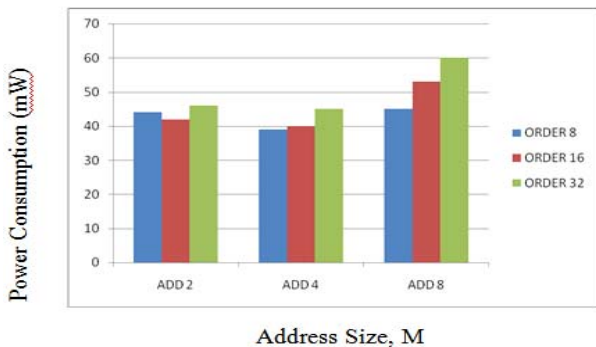


Figure 4. Plot of variation of Power Consumption with filter order for 1-D Decomposition method for $L = 8$.

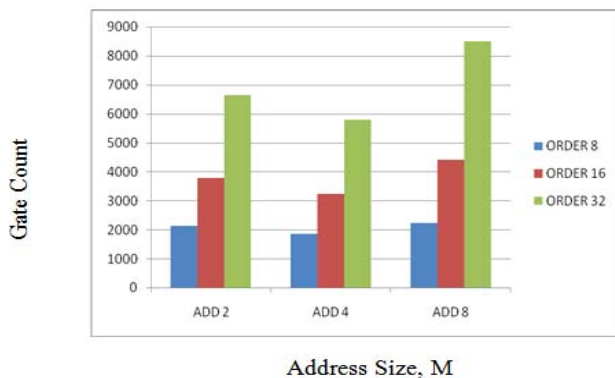


Figure 5. Plot of variation of Gate Count with filter order for 1-D Decomposition method for $L = 8$.

The above shown Table 2 is the comparison results of filters of different orders for 1-D systolic conventional method and 1-D systolic decomposition method. The decomposition method is better in all metrics for all values of N , as seen from the graphs shown in Figure 6 to 9. The synthesis tool used is Xilinx ISE 9.2i. The simulation tool used is ModelSim XE 6.2c. The target device selected is Virtex II Pro (XC2VP30).

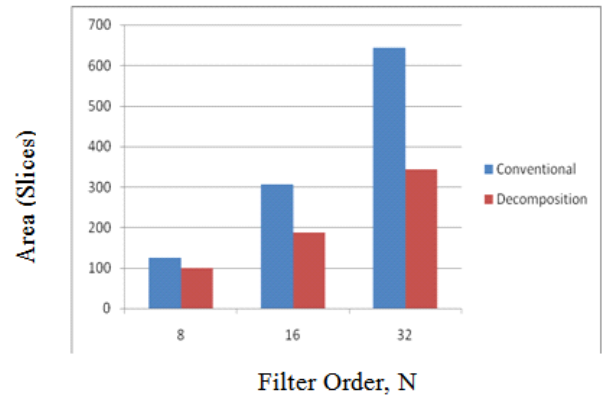


Figure 6. Plot of variation of Area with filter order for 1-D Conventional method and 1-D Decomposition method for $L = 8$.

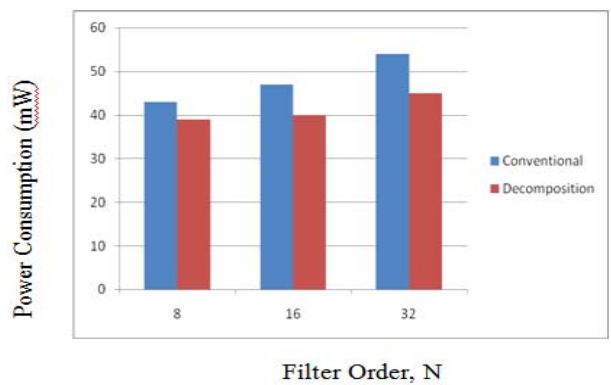


Figure 7. Plot of variation of Power Consumption with filter order for 1-D Conventional method and 1-D Decomposition method for $L = 8$.

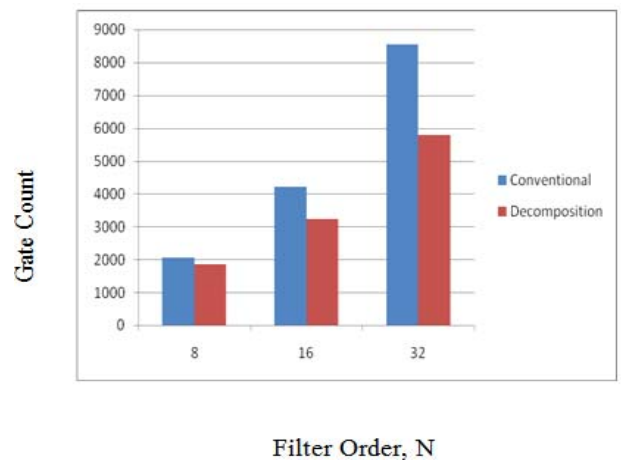


Figure 8. Plot of variation of Gate Count with filter order for 1-D Conventional method and 1-D Decomposition method for $L = 8$.

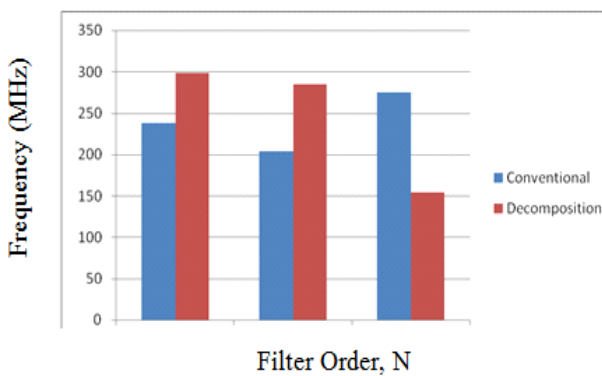


Figure 9. Plot of variation of Frequency with filter order for 1-D Conventional method and 1-D Decomposition method for $L = 8$.

TABLE III
COMPARISON OF THE EXISTING DAFIR SYSTEM GENERATOR BLOCK AND 1-D SYSTOLIC DECOMPOSITION METHOD WITH ADDRESS LENGTH $M = 4$ FOR FIR FILTER FOR WORD LENGTH $L = 8$:

FILTER ORDER (N)	EXISTING DAFIR SYSTEM GENERATOR BLOCK		1-D SYSTOLIC DA DECOMPOSITION METHOD	
	FREQUENCY (MHz)	POWER CONSUMPTION (mW)	FREQUENCY (MHz)	POWER CONSUMPTION (mW)
8	225.073	129	297.885	39
16	213.858	135	284.738	40
32	234.137	137	153.799	45

The address length M is taken to be four for the proposed implementation.

From the above shown Table 3 it is clear that the 1-D systolic decomposition method significantly outperforms the existing implementations in terms of two important key metrics, namely the frequency and power consumption for all the values of N .

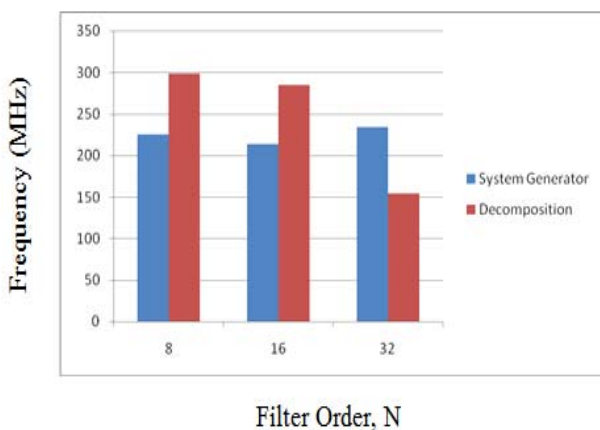


Figure 10. Plot of variation of Frequency with filter order for existing system generator block and 1-D Decomposition method for $L = 8$.

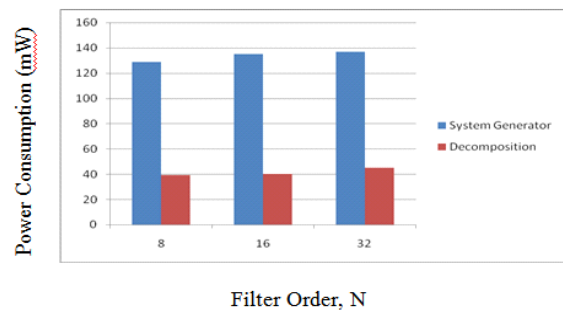


Figure 11. Plot of variation of Power Consumption with filter order for existing system generator block and 1-D Decomposition method for $L = 8$.

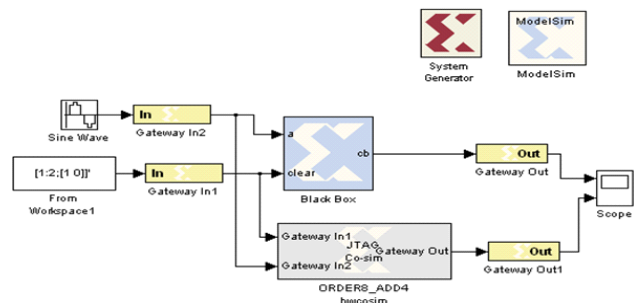


Figure 12. Hardware co-simulation of 1-D DA-based Decomposition method for $L = 8$.

CONCLUSION

The project presents hardware-efficient designs for computation of finite digital convolution by address decomposition of DA-based inner-product computation. The advantages of DA kind of implementation are its high usable frequency and minimum gate count. The main advantage is it overcomes the usage of multipliers. This method uses adders, LUTs and shift registers. The systolic decomposition scheme is found to offer a flexible choice of the address length of the lookup tables (LUT) for DA-based computation. The 1-D systolic array provides reduction in ROM size and the number of adders by several orders of magnitude compared to the conventional method.

REFERENCES

- [1] P. K. Meher, Shrutisagar Chandrasekaran and Abbas Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic", IEEE Trans. signal process., vol. 56, no. 7, pp. 3009-3017, July 2008.
- [2] B.K.Mohanty, P.K.meher "Cost effective novel flexible cell-level systolic architecture for high throughput implementation of 2D FIR filters" IEE 1996. Technical note.
- [3] P. K. Meher, "Hardware-efficient systolization of DA-based calculation of finite digital convolution", IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 707-711, Aug. 2006.
- [4] S. A. White, "Applications of the distributed arithmetic to digital signal processing: A tutorial review", IEEE ASSP Mag., vol. 6, no. 3, pp. 5-19, Jul. 1989.
- [5] H. T. Kung, "Why systolic architectures?", IEEE Computer, vol. 15, no. 1, pp. 37-45, Jan. 1982.
- [6] R. Wyrzykowski and S. Ovramenko, "Flexible systolic architecture for VLSI FIR filters", IEE Proceedings-E, vol.139,no.2,March1992